Leopard Server: Advanced setup, rsync backup and automated reporting

1.2

Setting up DNS 4
Testing DNS 5
Open Directory promotion 6
Verifying Open Directory connectivity 7
Automount creation 8
Nesting home directories 10
Redirecting folders with Workgroup Manager 13
Adding an OD Replica 15
Moving some of your network homes to new new OD Replica 16
Verifying permissions 16
How rsync works 17
KeyGen saves the day 17
Backing up from server to server 19
Removing cache files from home directories 19
Scripting the report 20
Backing up your Open Directory. 21
Launchd and Cron: Cron is gone 22
Summary 24

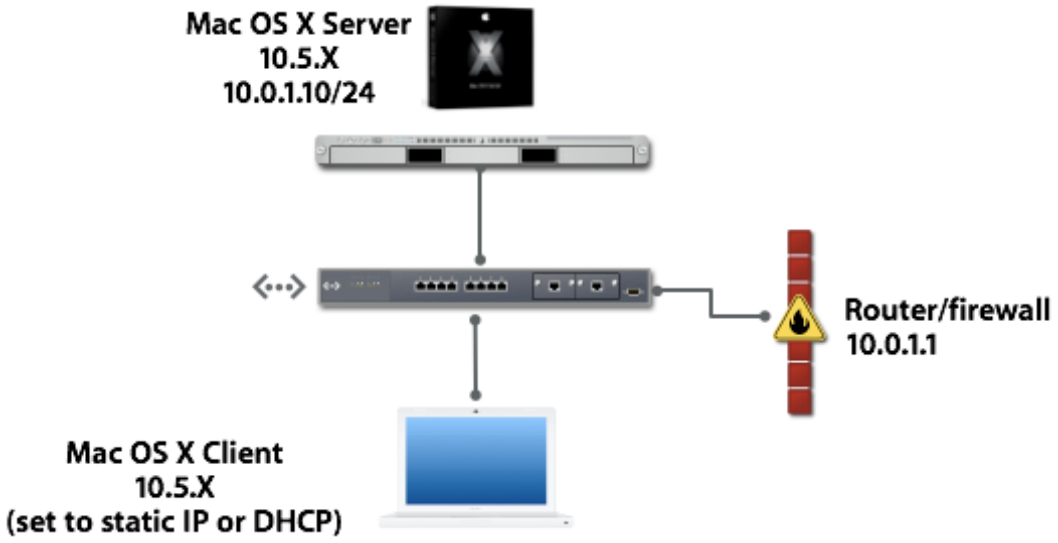
This document is simply an update of the Tiger Server Quickstart Guide posted on afp548.com back in August of 2005. You'll see a document that is quite similar, only updated where steps are different and screen shots have a new look. This document will follow the steps for an Advanced setup. For details regarding the differences between Standard, Workgroup and Advanced Server configurations, refer to page 21 of the document found at:
http://images.apple.com/server/macosx/docs/Getting_Started_v10.5.pdf

You've heard it plenty of times, and you're going to hear it again. It's all about Domain Name Service (DNS). It is the foundation of all directory services, and your world of user management will never run smoothly without it. Not only does Apple's Open Directory rely on it, but you would be ill-advised to ever attempt rolling out Active Directory, eDirectory or any other directory service without DNS. While this document will show you how to set up DNS on Leopard Server, you can utilize any standards-based DNS server. Just be sure the server provides both forward (A) and reverse (PTR) records, and you're good to go.

The environment

This setup consists of one server that provides DNS, Open Directory and AFP. The server is to be set up in **Advanced** Mode if you are to follow these steps. There is only one client computer in the setup, which is a 10.5 client computer. If you don't have another server or a router providing DHCP, you can add that service to this server if you wish. If you're simply testing Leopard Server and setting it up in a non-production environment, proper DHCP setup guidelines can be found in Apple's online server documentation. Specifically, the network services document is what you're looking for.

http://images.apple.com/server/macosx/docs/Network_Services_Admin_v10.5.pdf



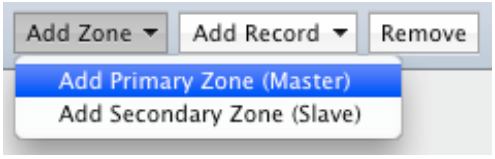
DNS Setup

Setting up DNS

If you don't have DNS in your school, here's a quick and simple guide to getting it up and running so that your server and clients can appropriately resolve names. This document shows steps from Server Admin. It is expected that you have gone through the installation and setup assistant at this point. During setup assistant for this installation, I set the server name as **odm.apple.edu** with the network settings as the following:

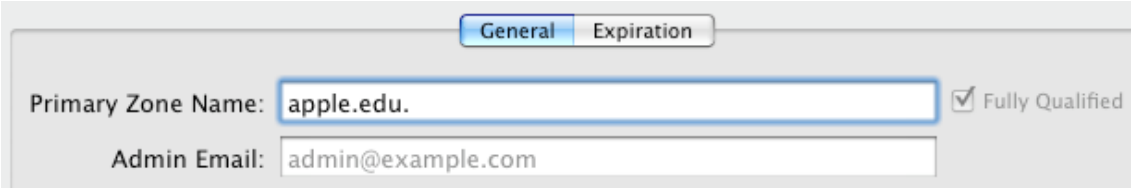
IP address:	**10.0.1.10**
Subnet mask:	**255.255.255.0**
Router:	**10.0.1.1**
DNS Primary:	**10.0.1.10** (should match IP address so that DNS setup goes smoothly)
DNS Secondary:	**10.0.1.1** (typically your home router so that you can run updates)

1. In **Server Admin**, click on the server you're going to configure.
2. Click on **Settings**, then **Services** and check the **DNS, AFP** and **Open Directory** services to enable them.
3. Click **Save**.
4. In the left pane of Server Admin, you'll see the DNS service. Select it and then click on **Zones**.
5. Click on the **Add Zone** menu and then **Add Primary Zone (Master)**.

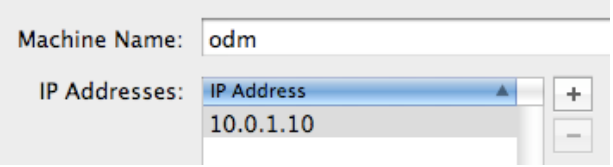


6. A zone based on **example.com** gets automatically created and is ready for you to edit it.

7. Select the **example.com** primary zone, and change the **Primary Zone Name** to **apple.edu**. Be sure to keep the trailing dot so that you're verifying that this is the fully qualified domain name.



8. Next, double-click on the first line of **Nameservers** and change **ns** to **odm** and click **Save**. Press return to set the nameserver and click **Save**. Notice that it changes the entry to a fully qualified domain name.
9. In the zone list above, click on the disclosure triangle to open the **apple.edu** zone. Select the **ns** Machine record.
10. Change the machine name to **odm**, leaving the **Fully Qualified** box **unchecked**.
11. Double-click on the IP address in the list (10.0.0.1) and change the IP to **10.0.1.10**, which is the IP address you set for your DNS server when using the setup assistant.



12. Click **Save**.
13. You'll see that a new reverse zone gets created by the name of 1.0.10.in-addr.arpa. That reflects the new IP address you put in that is in the 10.0.1.X range.
14. Select the **0.0.10.in-addr.arpa.** zone if it is still in the list and click **Remove**. It is unnecessary in this setup to have a reverse zone for an IP range that we're not using at this time.
15. Click **Save**.
16. In the lower-left of the Server Admin window, click **Start Service**.

Testing DNS

17. Be sure to add the DNS server into your TCP/IP settings, if you haven't already.
18. Once your zone is created, launch **Terminal** and type the following commands (with your IP and DNS information, of course):

```
host 10.0.1.10 <return>
host odm.apple.edu <return>
sudo changeip -checkhostname <return>
```

You should get feedback that looks similar to the results below, respectively.

```
10.1.0.10.in-addr.arpa domain name pointer odm.apple.edu
---
odm.apple.edu has address 10.0.1.10
---
Primary address  =   10.0.1.10
Current HostName =   odm.apple.edu
DNS HostName     =   odm.apple.edu
The names match. There is nothing to change.
```

19. Restart the server to ensure that DNS resolves and that your hostname is set properly. You probably don't need to restart, but we'll do this so that Open Directory promotion goes smoothly.

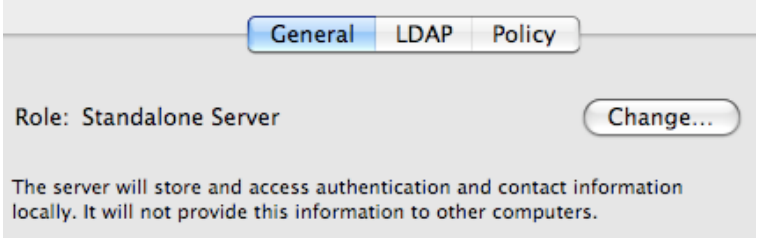
If you still need assistance, review the documentation for network services and/or call Apple Tech Support at (800) 800-2775.

http://images.apple.com/server/macosx/docs/Network_Services_Admin_v10.5.pdf

Open Directory

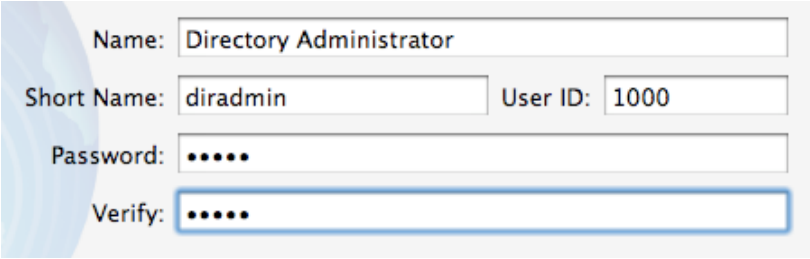
Open Directory promotion

1. Select the **Open Directory** service in Server Admin and choose **Settings**. You'll see that your server is currently set to Standalone.

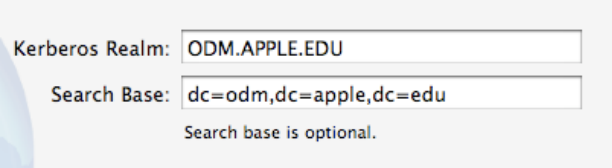

General
LDAP
Policy
Role: Standalone Server
Change...
The server will store and access authentication and contact information locally. It will not provide this information to other computers.

2. Click on the **Change...** button since we're going to promote this server to an Open Directory Master (ODM).
3. A new assistant included with Leopard Server will launch and walk you through the steps. Choose **Open Directory Master** and then the **Continue** button.

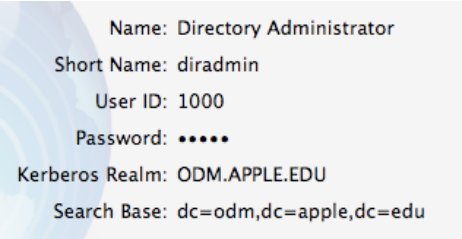
4. Set the Directory Administrator (diradmin) password and click **Continue**.



5. It is absolutely critical that the correct Kerberos and Search Base fields are set in this pane. Kerberos should show your FQDN in ALL caps and the search base should be your FQDN with **dc=** in between.



6. Confirm your settings and click **Continue**.



7. Click **Close** to finish the Assistant.

Verifying Open Directory connectivity

First, you must "bind" a client computer to your ODM. To do this, make sure DNS is resolving correctly on your client (both forward and reverse) by adding your DNS server to the list in the Network Preferences pane of System Preferences.

1. Launch the **Directory Utility** application (in /Applications/Utilities).
2. Unlock the application by clicking on the lock in the lower left-had corner of the window. Use your server admin account and password.
3. Click on the **+** button to add a server.

4. Enter the fully qualified domain name (FQDN) of your server, as shown below.

5. Once you click the **OK** button, you will be prompted to perform a Directory Bind. This is only required if you want to require a Directory Administrator password to unbind that computer from the OD Master. You can simply click continue at this screen while putting in no diradmin credentials.
6. Quit **Directory Utility**.
7. Launch **Terminal**.
8. Type the command **dscl** and press return.
9. At the > prompt, type **`cd /LDAPv3/FQDN/Users`** and press return. Note that you should replace **FQDN** with your server name, as demonstrated below with **odm.apple.edu**.

```
Entering interactive mode...
 > cd /LDAPv3/odm.apple.edu/Users/
/LDAPv3/odm.apple.edu/Users >
```

10. When you type **ls** and press return, it should give you a listing of all users in that LDAP directory. You should at least see the **diradmin** user.
11. If you have a list of users, then you have confirmed that you have successfully bound your client computer to your ODM. In the future, you can confirm connectivity by simply typing:

    `id diradmin`

    If it returns the correct uid, then you're bound to that directory.

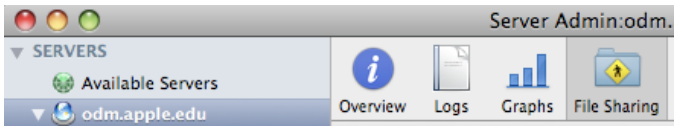
Network Home Directories

Automount creation

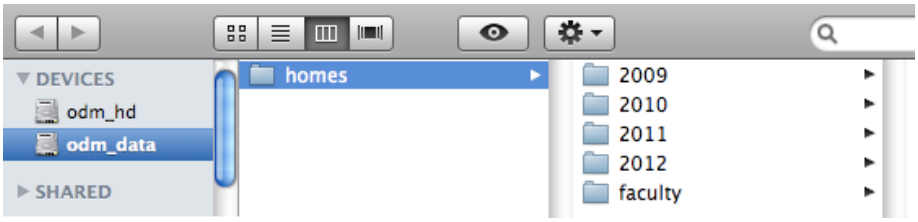
In order to allow login window authentication, and to mount the users home across the network, an automount needs to be created. It is very important to keep the number of automounts to a minimum. It is most often recommend to only utilize one (1) automount per server, if possible.

1. To create automounts, launch **Server Admin** on Leopard Server instead of Workgroup Manager. WGM was used to create automount records pre-Leopard.
2. Select the **AFP** service in the left pane of Server Admin and click on the **Start AFP** button in the lower left corner.

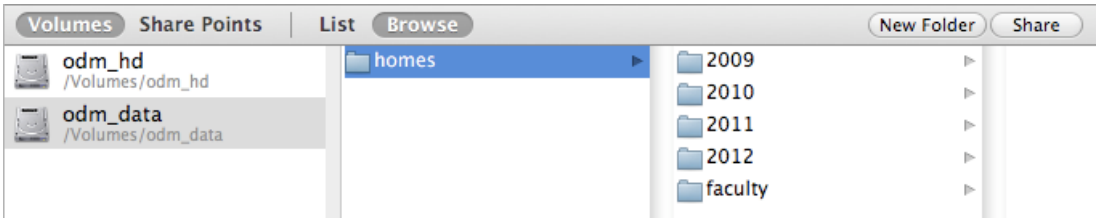
3. Highlight your ODM server (not any specific service) and click on **File Sharing** in the toolbar.



4. Click on **Share Points**. You'll see the list of default share points. If you don't intend on using them, highlight each one and unshare them by clicking on the **Unshare** button followed by the **Save** button.
5. Manually create the folder where all home directories will be stored. You can do this in Server Admin if you wish, but it might be faster for you to do this in the Finder if you're new to servers. To demonstrate best practices, it is shown here to put your home directories on a second drive that is separate from your OS boot drive or partition. The one folder is often called **homes** or the like, and then you can break the student and faculty home directories down by grade/class/alphabet within that folder however you choose.

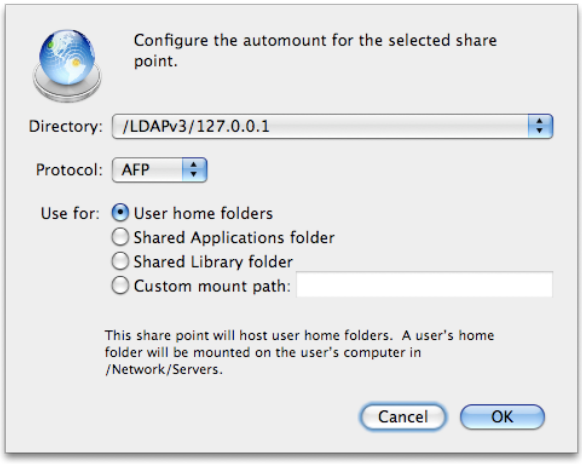


6. Return to **Server Admin** to edit File Sharing share points.
7. Click on the **Volumes** and **Browse** buttons. Navigate to the **homes** folder and click **Share**.



8. Click **Save**.
9. In the **Share Point** settings, check the **Enable Automount** button so that you can create network home directories for students and teachers to use.

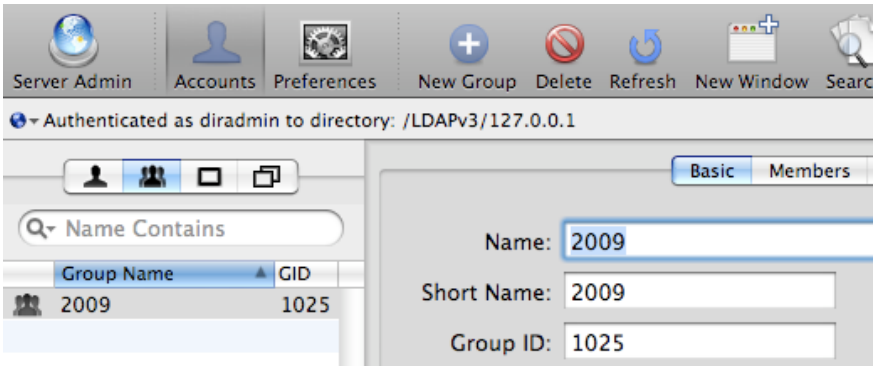
10. When prompted, choose the default settings of **LDAPv3**, **AFP** and **User home folders**. Click **OK**.


Configure the automount for the selected share point.
Directory: /LDAPv3/127.0.0.1
Protocol: AFP
Use for: User home folders
Shared Applications folder
Shared Library folder
Custom mount path:
This share point will host user home folders. A user's home folder will be mounted on the user's computer in /Network/Servers.
Cancel
OK

11. Type in your **diradmin** account and password to confirm.
12. Click **Save** in Server Admin.
13. Now that we have one network mount established, we can work on putting in test user accounts and establish how we're going to "nest" home directories within it.

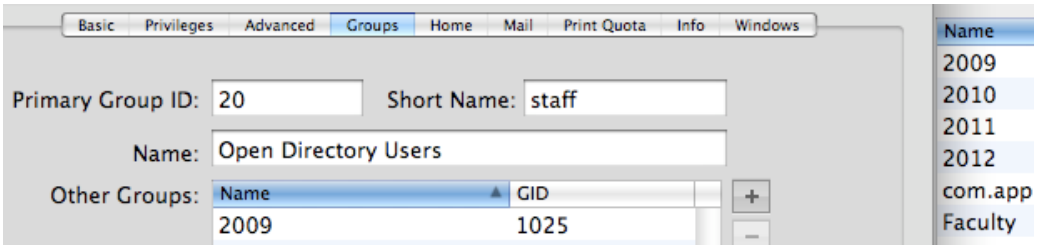
Nesting home directories

1. Launch **Workgroup Manager** (WGM) on the server and authenticate with the diradmin account.
2. Select the **Groups** tab and click on **New Group**.
3. Create a new group named **2009** and click **Save**.



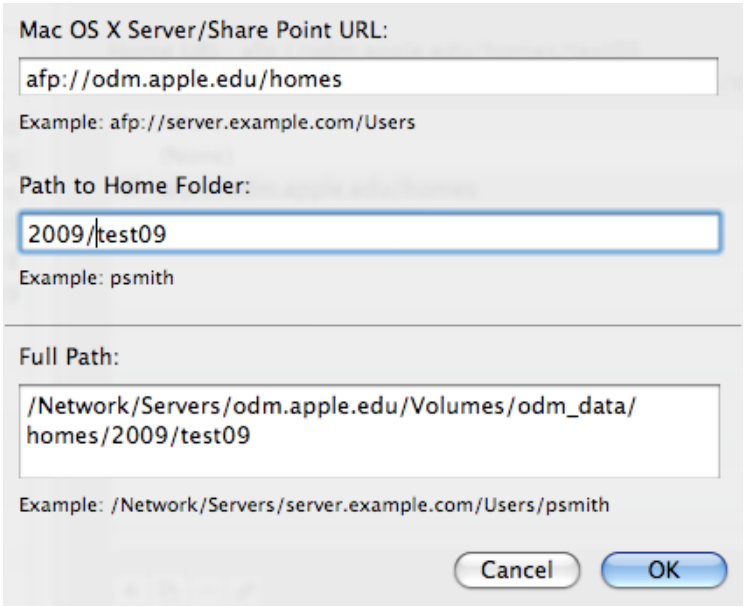
4. Repeat steps 1-3 to create groups named **2010**, **2011**, **2012** and **Faculty**. When you make the Faculty group, it's short name will be the same name in lowercase. Let it name it as such.
5. Select the **Users** tab and click on the **New User** button in the toolbar.
6. Give the account the name **Test 09**, which will automatically set the short name to **test09**. Set the password to whatever you like.

7. Select the **Groups** tab (within user settings) and click on the **+** sign.
8. Drag the **2009** group into the **Other groups** area and click **Save**. Be sure that the primary group remains the **staff** group, as shown below. Going forward, you can also double-click on the item to modify group membership in WGM.

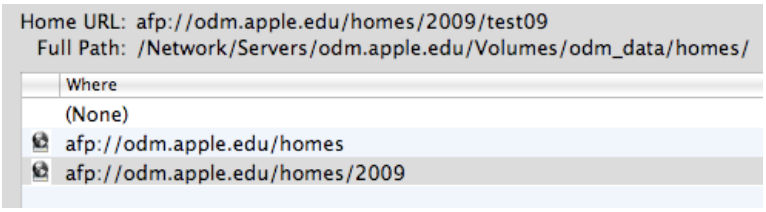


9. Repeat steps 5-8 to make the **Test 10**, **Test 11**, **Test 12** and **Test Faculty** user accounts. Be sure to make each user a member of their corresponding group.
10. In WGM, select your **Test 09** account and click on the **Home** tab. You should see a home location of **afp://odm.apple.edu/homes** and another as **(None)** as your two options. Remember that we made sub folders to better organize user home directories, so we'll need to modify this slightly.
11. Select the **afp://** location for the user's home, but don't hit the save button. Instead, click on the **Duplicate** button below the list of home folder options.

12. In the pane that appears, you simply need to type the name of the folder you created for that grade or class. In this case, the nested folder is titled **2009**. You only need to type that folder name in the **Path** field followed by a slash. It was previously **test09** and it is now set to **2009/test09**. You can also see that it updates the **Home** field below it with the new information.

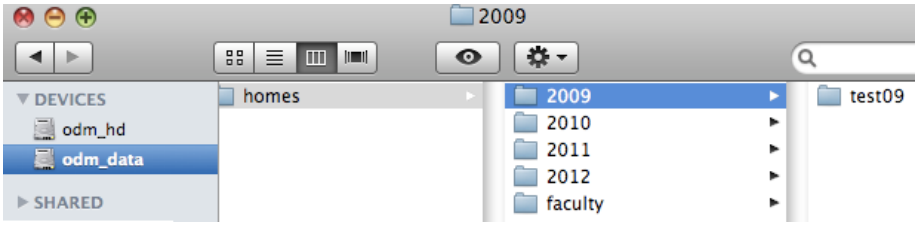


13. Click **OK**.
14. You'll see the new setting take place. The way this appears may make you think you just created a second network mount. No worries...you didn't.

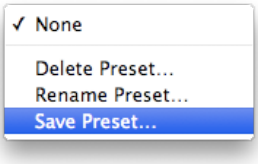


15. To test your new path, simply click on the **Create Home Now** button followed by **Save**.

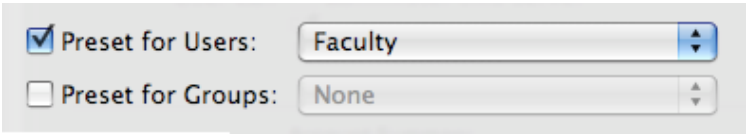
16. When you return to the Finder, you should see the new network home directory for the test09 user in the correct place.



17. Repeat steps 10-15 to test the **test10**, **test11**, **test12** and **testfaculty** accounts.
18. Before going any further, it is important to lay the foundation for more accurate ongoing directory administration. For each test account that you've confirmed for proper settings, group membership and home directory location, create a preset.
19. Highlight the **Test Faculty** account, and click on the Preset menu at the bottom of the WGM window.
20. Choose **Save Preset**.



21. Name the preset **Faculty** and click **OK**.
22. Under the **Server** menu, choose **Import**.
23. Without choosing a file to import, you can check the **Preset for Users** box and see that your **Faculty** preset is available.
24. Be sure that the Preset box is checked for Users, and choose your Faculty Preset.



25. If you do have a user import file, go ahead and do a test import. Upon importing, those new users will take on the settings that you gave your Test Faculty account. They will be members of the **Faculty** group, and their home folders (once created) will be created in the correct subfolder of your **homes** automount.

Reducing AFP load on your servers

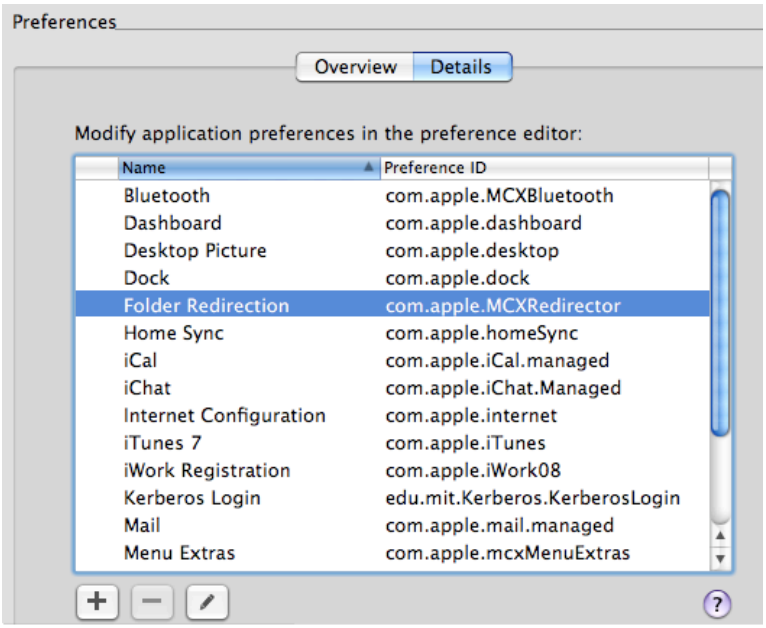
Redirecting folders with Workgroup Manager

Our good and trusted friend, Network Home Redirector, gets a well-deserved rest for the deployment of Leopard Server. For those that never deployed this tool, NHR proved invaluable to lab managers. It's primary purpose was to redirect files/folders from network home locations to the local HD, decreasing literally thousands (even millions) of files that would be constantly accessed across your network. It relieved your server from having to deal with all of these files (fonts, caches, etc.), and focus on user documents, which greatly increased it's performance and reliability. Here's to you, NHR!

This functionality is now built into Leopard Server. Let's take a look at how to use it.

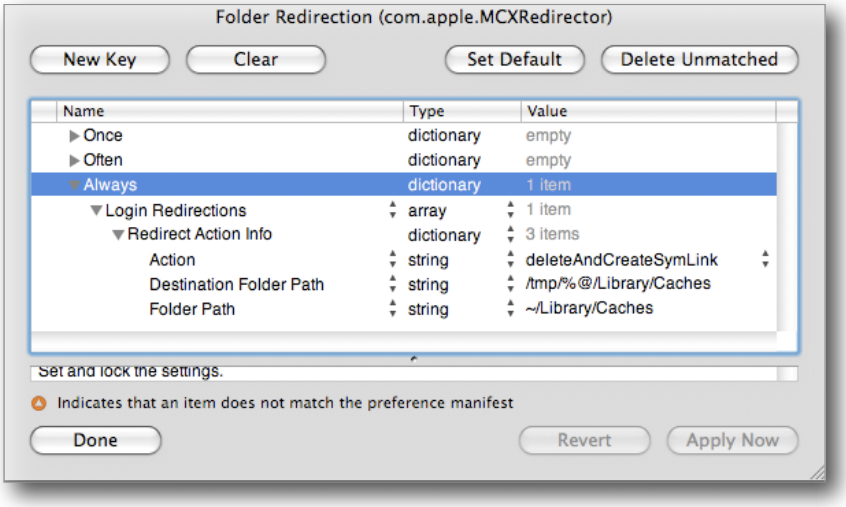
1. Open **WGM** and highlight the **2009** group.
2. Click on the **Preferences** button in the toolbar.
3. In the **Preferences** section of WGM, click on **Details**.
4. Click on the **+** button to add an application preference.
5. Navigate to **/System/Library/CoreServices**, choose **ManagedClient** and click **Add**.

6. In Leopard Server, this now easily puts all of the preference manifests in your hands to further manage your users. A big time-saver over Tiger Server.



7. Highlight the **Folder Redirection** field and click on the **Edit** button (pencil icon).
8. Click on the disclosure triangle next to the **Always** field and then select the line **Always**.
9. Click on the **New Key** button.
10. Select that entry named **New Item** and choose **Login Redirections**. Leave the type set to **Array**.
11. Leaving the **Login Redirections** highlighted, click on the disclosure triangle next to it.
12. Click the **New Key** button again. It will create a **Redirect Action Info** key. Leave the type set to **dictionary**.

13. Click on the triangle next to the **Redirect Action Info** entry, and you'll see the 3 included keys (**Action**, **Destination Folder Path** and **Folder Path**). You don't need to do anything else. This default policy will redirect user caches located in **~/Library/Caches** and redirect them to **/tmp**.



14. Click **Apply**, then **Done**.

15. Login as a member of that 2009 group and see the results.

You can enter multiple redirections for everything from Microsoft Office user data to fonts.

Adding an OD Replica

There are 4 roles of a Mac OS X server. They are Standalone, OD Master, OD Replica and Connected to a Directory. There are advantages to each of these roles, but if you are looking to add another server into your directory (that already has an OD Master), an OD Replica can be very beneficial. An OD Replica will store a read-only copy of the OD Master's database. This will allow you to share the load on your server, by allowing clients to split which server is authenticating them. The steps for setting up a replica are laid out very well in Apple's server documentation. Specifically, you want the Open Directory Administration guide, located here. http://images.apple.com/server/macosx/docs/Open_Directory_Admin_v10.5.pdf.

Moving some of your network homes to new new OD Replica

1. To move the home folders of the users, you can either use the command line (using a utility like **scp**, which stands for secure copy) or copy the homes to a firewire drive and then move them to the new server. If you are planning to use **scp**, the command would look something like the one below. *Don't execute this command until you read the next step.*

```
su <return>
```

```
scp -r -E -p/Volumes/odm_data/homes/2009 root@odr.apple.edu:/Volumes/
odr_data/homes/
```

(there is no space between "`odr.apple.edu:`" and "`/Volumes/odr_data/homes/`")

Heads up!

It is critical that you have the target folder already created on the target server that you're moving home folders to. Be sure that the **homes** folder is in the correct place, according to the path you have in your scp command above.

2. In **Server Admin** (on the ODR now), start the AFP service and follow the steps to create your new automount (it can also be named **homes**).
3. Once this is done, launch **WGM** and connect to the replica.
4. Highlight one of your users in WGM that just had their home folder moved to the ODR.
5. In the **Home** tab, create a new mount record just like it was done in step 14 of the previous section. Then, select all of the students from the same group, and set their homes to the new location.

Verifying permissions

The great thing about using scp is that you added the -p flag, which retains the permissions of the folders and files. These steps used to verify permissions is only needed if you use scp, ditto or rsync and do not retain permissions.

6. The final step in migrating users to a different server is to run Passenger (or your own script) to change the permissions to the correct user/group/everyone on all the files and folders that have been moved from the odm to the odr.
7. If you are using Passenger, you would want your screen to look like this:

```
/Volumes/odr_data/homes/2009/<username>
Owner - <username> - R/W
Group - <administration> - R/W
```

(Administration is a group that we recommend creating in WGM, that has the local server admin account and any other account that the school wants to be able to have R/W to the student's directories). It is optional, though.

```
Everyone - None
```

How rsync can handle your backups

How rsync works

Rsync is an open source utility that provides fast incremental file transfers for backups, which comes installed on every machine Apple ships. With rsync, the program does a full backup the first time through. Then, on subsequent backups, rsync will simply update the backup with the changes that have occurred since the last backup. One benefit of rsync is that it can back up data to a locally attached device, as well as remote hosts on the network (via ssh, for example). If you would simply like to copy files from every user's Documents folder on the server, to another drive, this could be done very quickly and efficiently. In the example below, every user's Documents folder will be copied (the Homes of all my users are in **/Volumes/odm_data/homes/**) to the backup drive located at **/Volumes/Backup/**.

```
rsync -a -R /Volumes/odm_data/homes/*/*/Documents /Volumes/Backup/
```

Every time this command runs, the files that have been modified in the user's Documents folder will be copied to the folder **Backup**, but any files that have been deleted from their Documents folder will not be deleted off the backup copy. If you would like to delete the files on the backup so that it truly matches the file count of your home folders, simply add the **--delete** command like this:

```
rsync -a -R --delete /Volumes/odm_data/homes/*/*/Documents /Volumes/Backup/
```

If you are looking to take this to the next level, you could have this backup running to a remote server or client somewhere else on your network, so that you would have a true off-site backup. To do this, you will first have to set up a key on both your server and the targeted backup server, so that when it goes to run the backup job via ssh, the server it is connecting to doesn't have to ask for a password.

KeyGen saves the day

The most common authentication method used by ssh (secure shell) is a username and password. The experience is very similar to what you've seen before. However, when one machine connects to another via ssh, the ssh client and the sshd process on the server jointly determine a private session key to keep the session, well, private. In this situation, however, we don't want our backup to have to wait for someone to type in a password. For this purpose, you can use ssh key pair authentication. When key pair authentication is enabled, the ssh server looks into a user's ssh directory, finds the public key, and uses it to create a challenge for the client. The client in turn verifies his identity (answers the challenge) using the private portion of the key. You can create ssh key pairs with the ssh-keygen command.

It's also important to note that you can have any user on the server benefit from KeyGen. Although these steps utilize the root user, you can use any server administration account. If you choose to do so, be sure that the specific user you utilize has ownership to the all of the files you create in the steps below.

1. Open **Terminal** and type:

   ```
   su
   ```

   This will log you in as the root user, and will ask you for the root password (which, by default, is the first local administrator account password you created). Type this command to start creating the keys:

   ```
   ssh-keygen -t dsa
   ```

   It will ask you where you want to save the key, which by default will be in your home folder, in a folder called **.ssh**, and will be named **id_dsa**. It will then ask you for a passphrase.

2. Simply hit return, twice to enter no password and no verify password. When it is done, it will have created both the public and private key pairs in your **~/.ssh** directory. The files are:

   id_dsa This is your private key, should only be readable by the owner and kept very secure. Some users even place this file on an encrypted disk image.

   id_dsa.pub This is the public portion of the key, which does not have to be kept secure. This file will be copied to every server you want to SSH to, and its contents added to your ~/.ssh/authorized_keys file. The authorized_keys file contains a list of all public keys for that account.

3. Copy your public key to your server using the scp command:

   ```
   scp ~/.ssh/id_dsa.pub root@server.yourdistrict.org:
   ```

4. Next, you will need to ssh into your server via this command:

   ```
   ssh root@server.yourdistrict.org
   ```

5. Once you are logged into your server via ssh, run this comand:

   ```
   mkdir ~/.ssh
   cat ~/id_dsa.pub >> ~/.ssh/authorized_keys
   exit
   ```

Subsequent ssh connections to that server will not require a password.

Backing up from server to server

Once this is setup, you can send a command like this to backup every user's Documents folder on the server (just like in the previous example) to a remote server. For this example, the remote server address is 10.0.1.3, the username to connect to the server via ssh is root and the location of the backups will be in /Volumes/odr_data/Backup/.

```
rsync -a -R /Volumes/odm_data/homes/*/*/Documents root@10.0.1.3:/
Volumes/odr_data/Backup/
```

And just like before, if you want to delete the files/folders that have been deleted from the user's documents folder, simply add the **--delete** syntax to this command. Rsync is lightweight, free, very fast and can be added to launchd, so that this can be fully automated. The first backup will take quite a while to run (depending on the size of the directories to backup and your network bandwidth), but after that, this off-site backup could run fairly quickly (again, based on size and speed).

Removing cache files from home directories

The **find** command can save any server administrator a tremendous amount of time. It can comb through all user home directories and delete whatever you ask it to. The command below will empty the trash of all users within the "homes" automount on the OD Master.

```
find /Volumes/odm_data/homes/*/*/.Trash/* -exec rm -rf {} \;
```

Having your servers email reports

Scripting the report

The script shown below will email a report that includes disk free space, current AFP connections and illegal attempts to authenticate via ssh (both internally and externally).

```
#!/bin/sh
PATH=/bin:/usr/bin:/sbin:/usr/sbin export PATH

#Path to logs
LOGS="/tmp/Daily.txt"

echo "From: Server Daily Report <username@emaildomain.org>" > $LOGS
echo "To: username@emaildomain.org" >> $LOGS
echo "Subject: Daily Report" - `date` >> $LOGS
echo "UPTIME" >> $LOGS
echo "------" >> $LOGS
echo `uptime` >> $LOGS
echo "------" >> $LOGS
echo " " >> $LOGS
echo "FREE SPACE" >> $LOGS
echo "------" >> $LOGS
df -klh >> $LOGS
echo "------" >> $LOGS
echo " "
echo "CURRENTLY ESTABLISHED CONNECTIONS" >> $LOGS
echo "------" >> $LOGS
netstat -an | grep ESTABLISHED | grep 10.0.1.2 >> $LOGS
echo "------" >> $LOGS
echo " " >> $LOGS
echo "ILLEGAL ATTEMPTS" >> $LOGS
echo "------" >> $LOGS
grep -i "failed" /var/log/secure.log >> $LOGS
echo "------" >> $LOGS
echo " " >> $LOGS
echo "SUCCESSFUL INTERNAL ATTEMPTS" >> $LOGS
echo "------" >> $LOGS
cat /var/log/secure.log |grep -i "Accepted publickey" | grep -i
"10.0.1" >> $LOGS
echo "------" >> $LOGS
echo " " >> $LOGS
echo "SUCCESSFUL EXTERNAL ATTEMPTS" >> $LOGS
echo "------" >> $LOGS
cat /var/log/secure.log |grep -i "Accepted" | grep -v "10.0.1" >>
$LOGS
echo "------" >> $LOGS
echo " " >> $LOGS

cat $LOGS | sendmail -f username@emaildomain.org -t
```

Automating Open Directory Backups

Backing up your Open Directory.

As demonstrated earlier, you will first want to leverage launchd and set up a task that will run on a repeating schedule. Once that is set, this script will back up Open Directory (just like Server Admin does, in Open Directory -> Archive). It will back it up on a schedule that will remove the sparseimage, after it has been sitting on the server for 7 days (so you will have 7 full days of backups - which you can change to let it stay on the server for as many days as you want).

```
#!/bin/sh
#set -xv; exec 1>>/tmp/out 2>&1
PATH=/bin:/usr/bin:/sbin:/usr/sbin export PATH

FILE="/tmp/sacommands.txt"
LOCATION=/Backup/Server/`date "+%Y_%m_%d"`
LOGS="/tmp/logs.txt"

mkdir -p /Backup/Server
find /Backup/Server* -mtime +6 -exec rm -rf {} \;
echo "dirserv:backupArchiveParams:archivePassword = YOURPASSWORDHERE"
> $FILE
echo "dirserv:backupArchiveParams:archivePath = $LOCATION" >> $FILE
echo "dirserv:command = backupArchive" >> $FILE
serveradmin command < $FILE
echo "To: username@emaildomain.org" > $LOGS
echo "From: Server Daily Report <username@emaildomain.org>" >> $LOGS
echo "Subject: Daily ODM Backup Report" - `date` >> $LOGS
echo "The Open Directory Master at `hostname` has been successfully
backed up. It's location is $LOCATION and will be available for seven
days. After seven days, the archive will be deleted." >> $LOGS
cat $LOGS | sendmail -f username@emaildomain.org -t
rm -rf $FILE
rm -rf $LOGS
```

Launchd and Cron: Cron is gone

Cron is a UNIX process that runs by default on all computers running Mac OS X 10.4 or Mac OS X Server 10.4. It's only job is to check at the top of every minute, to see if any jobs need to be executed. In previous documents, we always scheduled rsync backups or automated daily reports with cron. As of the Leopard (10.5) version of Mac OS X and Mac OS X Server, doing a clean install of these operating systems will ensure that the crontab file is missing from the **`/etc`** directory. The reason for this is because **launchd** is the new task scheduler/manager that replaces cron. If you do an upgrade from 10.4 to 10.5 (client or server), cron will still be there. It's time to do a crash course on launchd, because that's what Apple is using now. Hopefully these steps will get you over the hump and onto creating your first scheduled job.

Creating a job using launchd

We'll be using Lingon to create launchd jobs. Lingon is an open source tool that works very well for seasoned admins as well as those setting scheduled jobs for the first time. These steps assume that you've created actual script files that you can use with Lingon. The best way to do that is to use a tool like TextWrangler or TextMate, and simply copy/paste the text from the above sample scripts. Of course, you'll need to customize them for your environment.

1. Download and install **TextWrangler** from http://www.barebones.com if you don't have a current favorite text editor. You can use another text editor such as TextMate, BBEdit or SubethaEdit if you choose. TextWrangler is shown in the example because it does the job and it's free.
2. **Copy** the text from one of the scripts examples on the previous 2 pages and **paste** it into your a new document in the editor of your choice. Customize the script for your environment.
3. Save it as whatever name you like (we'll use **daily_report.sh** as an example) and be sure to save it in **/Library/Scripts** on your server (not the client workstation).
4. Launch **Terminal** on your server (or ssh into it if you like) and type the following commands, each line followed by the return key:

   `sudo chmod 700 /Library/Scripts/daily_report.sh` (use the file name you chose earlier)
   `sudo chown root /Library/Scripts/daily_report.sh`

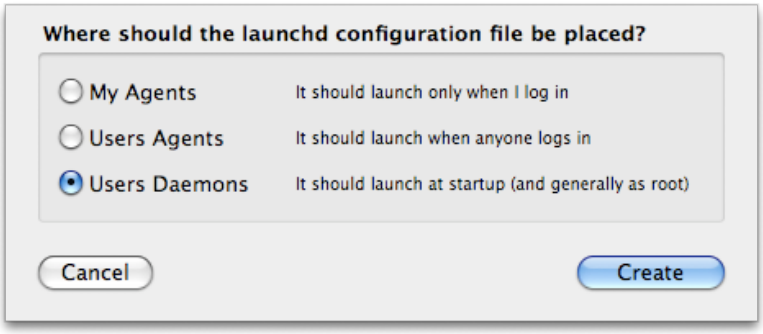
   These two steps simply make the script executable (**`700`**), and change the owner to the root user (**`chown root`**).

5. Download and install Lingon on your Leopard Server from http://lingon.sourceforge.net/.

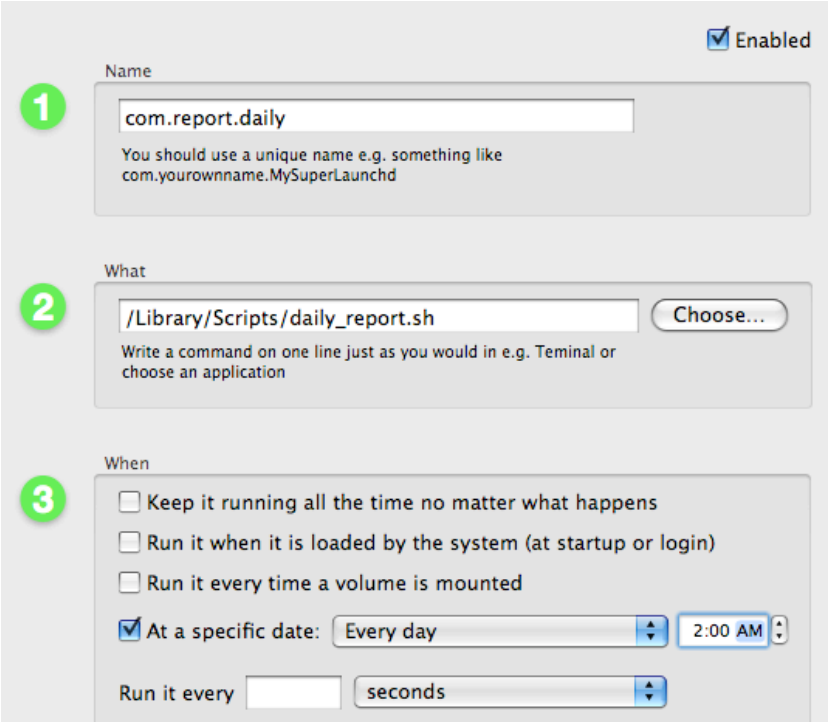
Lingon

6. After copying the application to the Applications folder, launch **Lingon**.

7. Choose **File** -> **New** and decide what kind of agent you want. For the sake of this document, create a **User Daemon**.


Where should the launchd configuration file be placed?
My Agents
It should launch only when I log in
Users Agents
It should launch when anyone logs in
Users Daemons
It should launch at startup (and generally as root)
Cancel
Create

8. Give it a name so that it can be identified. If this were a daily reporting script for your server, you could name it **com.report.daily**.
9. On the next line, select the **Choose** button and browse to find the script you want to run.
10. Choose when you want the script to run and save the agent using the **Save** button in the upper right of the window.
11. You will be prompted for your administrative username and password. You should also be prompted for a restart, so do that as well. Once the launchd job is saved, the script will run when the criteria in section 3 of this screen shot are met.


Enabled
Name
1
com.report.daily
You should use a unique name e.g. something like com.yourownname.MySuperLaunchd
What
2
/Library/Scripts/daily_report.sh
Choose...
Write a command on one line just as you would in e.g. Teminal or choose an application
When
3
Keep it running all the time no matter what happens
Run it when it is loaded by the system (at startup or login)
Run it every time a volume is mounted
At a specific date:
Every day
2:00 AM
Run it every
seconds

12. In the Finder (still on your server), navigate to **/Library/LaunchDaemons** and you'll see your newly created launchd job. Open it up in a text editor and see the contents. The contents of the file should look similar to what's below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>com.report.daily</string>
        <key>ProgramArguments</key>
        <array>
             <string>/Library/Scripts/daily_report.sh</string>
        </array>
        <key>StartCalendarInterval</key>
        <dict>
             <key>Hour</key>
             <integer>2</integer>
             <key>Minute</key>
             <integer>0</integer>
        </dict>
</dict>
</plist>
```

13. If you need to make changes to the launchd job, feel free to edit the job itself with a text editor (GUI or command line). You can also simply delete it and create a new one with Lingon.

Summary

While this document may shed some light on the topics of initial setup, backup, scripting and automated reporting, it is still only scratching the surface of these strategies. Apple's server documentation, online Knowledge Base (KBase) articles and enterprise support team are still the best resources for having your open directory infrastructure running smoothly.

http://www.apple.com/server/documentation
http://www.apple.com/support/products/macosxserver_sw_supt.html

In addition, many other great articles, links and white-papers are hosted by the following sites.

http://www.afp548.com
http://www.macenterprise.org

Special thanks to Jeff Ochsner for helping test the processes and steps outlined in this document. Comments and corrections for this document can be sent to me at carson@mac.com.